Attorney Docket No. 81862.P187 *PATENT*

UNITED STATES PATENT APPLICATION

FOR

DELIVERY OF A SERVICE PROGRAM TO A DIGITAL SIGNAL PROCESSOR WITHIN A MULTISERVICE PROCESSING SYSTEM

Inventors:
John Baggs
Louis Couture

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026
(408) 720-8300

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EL 617 210 129 US

Date of Deposit December 22, 2000

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

Tina Domingo
(Typed or printed name of person mailing paper or fee)

12-22-2000
(Signature of person mailing paper or fee) Date

DELIVERY OF A SERVICE PROGRAM TO A DIGITAL SIGNAL PROCESSOR WITHIN A MULTISERVICE PROCESSING SYSTEM

CROSS REFERENCE TO RELATED APPLICATION

The invention is related to "A method and apparatus for supporting multiservice digital signal processing applications," invented by Louis Couture, filed March 26, 1998, U.S. Application No. 09/049,288.

FIELD OF THE INVENTION

The invention is related to the field of digital signal processing.

BACKGROUND OF THE INVENTION

In multiservice call processing applications (e.g., voice, fax, modem, and video communications), a matrix of processors are typically required to process a number of channels to provide multiple services on one hardware platform. Ideally, a service provider using such equipment would want to be able to service any type of incoming call with the same hardware box on a mixed, random, and dynamic basis. Different firmware images thus need to be loaded and executed by the processors to accommodate the various processing requirements associated with these different services.

A typical solution is to store all of the possible algorithms that may be needed in the memory of each processor. However, this solution is not economically practical, because of the large memory requirements for this solution.

SUMMARY AND OBJECTS OF THE INVENTION

A method including determining whether a digital signal processor needs a service program stored in a juke box overlay memory, and delivering the service program to the digital signal processor from the juke box overlay memory over a host port interface bus is disclosed.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements, and in which:

Figure 1 is a communications system including a multiservice processing system of one embodiment.

Figure 2 shows an embodiment of a multiservice processing system.

Figure 3 shows one embodiment of a method for delivering a service program to a digital signal processor within a multiservice processing system.

DETAILED DESCRIPTION

A method for delivery of a juke box overlay program via a processor host port from a juke box overlay memory that stores a library of programs to an internal program memory of a digital signal processor is disclosed. The overlay program can be used to replace or modify a program that is stored in the program memory of the digital signal processor. The programs in the library are used to provide different services to a multiple service system. In one embodiment, a packet pump manager tells a digital signal processor to process packets using a given service. The digital signal processor indicates that it does not have the program for the given service. A juke box overlay manager then causes the juke box overlay manager to download the program to the internal memory of the digital signal processor. The digital signal processor then executes the program and provides the required service to the packets that it processes. Thus, the apparatus can provide service programs from the library to one or more processors on a mixed, random, and dynamic basis.

Figure 1 is a communications system 100 including a multiservice processing system 102 of one embodiment. The multiservice processing system 102 receives data from a public switched telephone network (PSTN) 104 over a multiplexed line 106. The data is received as pulse coded modulation (PCM) data streams, but the embodiment is not so limited. In one embodiment, the multiplexed line 106 comprises 24 multiplexed data lines, but the embodiment is not so limited. The multiplexed line 106 may be a T1 data line, a T3 data line, or an E1 data line, but the embodiment is not so limited. The data types 108-116 received over the PSTN 104 comprise, but are not limited to, facsimile data 108, modem data 110, video data 112, audio data 114, and voice data 116, for example telephone data. The lines 128-136 over which the data 108-116, respectively, is provided may be multiplexed data lines, but the embodiment is not so limited. The multiservice processing system 102 processes the received PCM data and generates packets or cells comprising the processed data. The packets or cells are provided to an IP network 118 for transmission.

The multiservice processing system 102 also receives packetized data from the internet protocol (IP) network 118. The data types received in packets or cells over the IP network 118 comprise, but are not limited to, facsimile data, modem data, video data, audio data, and voice data, for example telephone data. The multiservice processing system 100 unpacks the packetized data and generates PCM data streams comprising the data. The PCM data streams are provided to the PSTN 104 using the multiplexed line 106. The PSTN 104 distributes the PCM data streams to individual subscribers 108-116 of particular destinations using data lines 128-136. The data lines 128-136 may be multiplexed data lines, but the embodiment is not so limited.

Figure 2 shows an embodiment of multiservice processing system 102. Field processor gate array (FPGA) packet pump 240 sends and receives the

PCM data streams to and from the PSTN. Packet pump 240 sends the data streams to and from appropriate digital signal processors (DSP) 250 through HPI bus 270 using a packet pump host port interface (HPI) bus manager 245, which may be implemented as a state machine for example. Each DSP 250 converts PCM data streams received from the packet pump into IP packets, and places the packets onto the IP network using a firmware or software program stored in the DSP internal memory. Each DSP 250 also converts packets received from the IP network into PCM data streams using the stored program, and sends the data streams to packet pump 240. The program may be placed in the DSP internal memory by juke box overlay memory 250. Juke box overlay memory 250 stores a library of software programs, and can download the library of software algorithms to the DSPs 250 through the HPI bus. The juke box overlay memory may be a static random access memory (SRAM), for example. The memory 260 is a "juke box" overlay memory because the memory 250 stores several service programs, thus functioning as a "juke box."

The program stored in each DSP internal memory is used to provide a service to the data packet or data stream that is processed by the DSP. For example, a fax program can be used to convert a fax data stream into a fax IP packet. The packet pump tells the DSP which service to provide to a given data stream or data packet. If the DSP has the corresponding service program stored in its internal memory, the DSP processes the data stream or data packet by executing the stored program. If the corresponding service program is not stored in the internal memory of the DSP, then the DSP indicates that it does not have the service program. A juke box overlay manager 245, which may be part of HPI bus manager 243 in the packet pump, then causes the juke box overlay memory to download the needed service program to the internal memory of the DSP. The DSP then processes the packet or data stream using the downloaded service program.

Figure 3 shows one embodiment of a method for delivering a service program from the juke box overlay memory to the DSP internal memory using a HPI bus within a multiprocessing system. The service programs for the multiservice processing system are loaded into the juke box overlay memory when the multiservice processing system is initialized, 310. A juke box overlay manager in the packet pump is periodically invoked, 320. The juke box overlay manager may be invoked by the HPI bus manager in the packet pump.

The juke box overlay manager determines whether one or more DSPs need a service program. The juke box overlay manager also determines which service program or programs are needed, 330. For example, each DSP may indicate that it needs a given service program by setting a flag in the internal memory of the DSP. The juke box overlay manager then can read the flag to determine that the DSP needs a program, and to determine which program is needed by the DSP.

The juke box overlay manager then schedules the delivery of the needed service program or programs to the corresponding DSPs, 340. For example, the juke box overlay manager may prioritize the delivery of several needed programs by determining the number of DSPs that need each requested program. The program that is needed by the most DSPs may then be delivered first. Scheduling the delivery of a program may also include determining the size of the needed program, determining where the needed program is located in the juke box overlay memory, and determining the memory addresses of each DSP internal memory that will receive the program. For example, the programs that are stored in the library can be indexed by size, delivery location, and storage offset, to allow a state machine to deliver a given program without requiring an additional processor.

The juke box overlay manager then causes the juke box overlay memory to download the needed program directly into the internal memory of one or more DSPs through the HPI bus, 350. If several DSPs need the service program, the service program may be simultaneously downloaded in parallel to all DSPs that need the program. If a subsequent DSP requests the program while the program is being downloaded, the DSP can start receiving the download. The program code that was not initially delivered to the subsequent DSP can be downloaded at a later time. Also, other DSPs can place a request for other programs in a request queue while a given program is being downloaded.

After the program is downloaded, the juke box overlay manager sends a signal indicating that the download is complete, 360. The DSPs that received the downloaded service program in their internal memory then process data packets and data streams by executing the downloaded program. The juke box overlay manager may then schedule another program to be downloaded, or may return control of the HPI bus to the HPI bus manager if a given amount of time has expired. The HPI bus manager then resumes sending and receiving packets from and to the FPGA, and then subsequently invokes the juke box overlay manager, as discussed above.

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes can be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.